

Sistemi Operativi L-A
Prof. Anna Ciampolini
Prova Scritta di Lunedì 10 luglio 2006

Tempo a disposizione: 3 ore

Esercizio 1 [punti 12] – Progetto UNIX

Si realizzi un programma, che, utilizzando le *system call* del sistema operativo UNIX, soddisfi le seguenti specifiche:

Sintassi di invocazione:

esame COM1 COM2 N

Significato degli argomenti:

- **esame** è il nome del file eseguibile associato al programma.
- N è un valore intero positivo.
- **COM1** e **COM2** sono stringhe che rappresentano nomi di file eseguibili (per semplicità, si supponga che il direttorio di appartenenza dei file COM1 e COM2 sia nel PATH)

Specifiche:

- Il processo iniziale (P_0) deve creare 2 processi figli P_1 e P_2
- Il processo P_1 , una volta creato, dovrà eseguire il comando COM1
- Il processo P_2 , una volta creato, dovrà eseguire il comando COM2
- Il padre P_0 , una volta creati i figli P_1 e P_2 , si pone in attesa della terminazione di essi; se uno dei due figli termina prima di N secondi (misurati dall'istante immediatamente successivo alla creazione di P_2) P_0 dovrà provocare la terminazione forzata dell'altro figlio.
- In ogni caso: P_0 dovrà stampare lo stato di terminazione di ciascun figlio sullo standard output, e successivamente terminare.

Esercizio 2 [punti 18] – Monitor

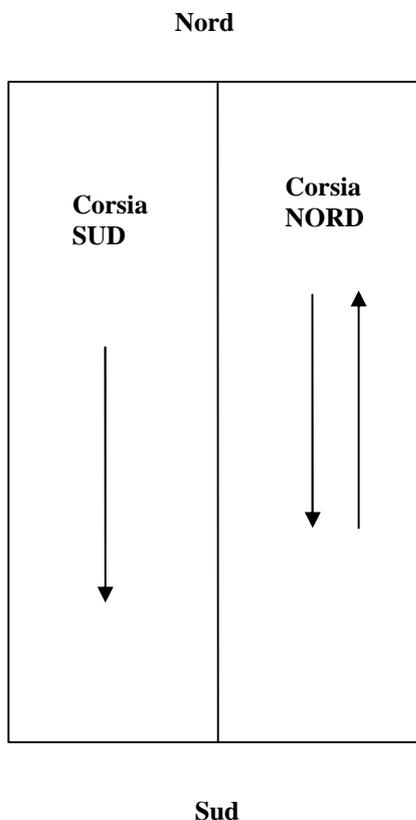
Si consideri un tratto di autostrada, nel quale si stanno svolgendo lavori di manutenzione. Per questo motivo, il tratto considerato è soggetto a **particolari condizioni di uso**.

In particolare, sono presenti una *corsia Nord* e una *corsia Sud*, ciascuna a capacità limitata MAX (che esprime il massimo numero di veicoli nella corsia).

Nel tratto considerato il traffico è molto più intenso nella direzione NS (Nord->Sud) rispetto alla direzione SN (Sud->Nord).

Pertanto, allo scopo di ridurre i problemi di congestionamento, la gestione del tratto autostradale avviene secondo la seguente politica :

- la corsia Sud viene percorsa a **senso unico nella sola direzione NS**;
- la corsia Nord viene utilizzata a **senso unico alternato** (cioè: non può essere contemporaneamente percorsa da veicoli in direzioni opposte) secondo la seguente politica:
 - Quando la corsia Sud è piena, i veicoli provenienti da nord vengono deviati nella corsia Nord;
 - Nell'accesso alla corsia Nord, si dà la **precedenza ai veicoli provenienti da sud**.



Si modellino i veicoli mediante thread concorrenti e si realizzi¹ una politica di sincronizzazione tra i thread basata sul concetto di monitor che tenga conto dei vincoli dati.

¹ Realizzazione: il candidato può scegliere tra il linguaggio C/libreria pthread, o il linguaggio Java.